



Rovigatti, L., Russo, J., & Romano, F. (2018). How to simulate patchy particles. *European Physical Journal E*, 41(59).
<https://doi.org/10.1140/epje/i2018-11667-x>

Peer reviewed version

Link to published version (if available):
[10.1140/epje/i2018-11667-x](https://doi.org/10.1140/epje/i2018-11667-x)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via SPRINGER at <https://link.springer.com/article/10.1140%2Fepje%2Fi2018-11667-x>. Please refer to any applicable terms of use of the publisher

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

How to Simulate Patchy Particles

Lorenzo Rovigatti*

*CNR-ISC, Uos Sapienza, Piazzale A. Moro 2, 00185 Roma, Italy and
Dipartimento di Fisica, Sapienza Università di Roma, Piazzale A. Moro 2, 00185 Roma, Italy*

John Russo

School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom[†]

Flavio Romano

*Dipartimento di Scienze Molecolari e Nanosistemi,
Università Ca' Foscari di Venezia, Via Torino 155, 30172 Venezia Mestre, Italy[‡]*

(Dated: April 17, 2018)

Patchy particles is the name given to a large class of systems of mesoscopic particles characterized by a repulsive core and a discrete number of short-range and highly directional interaction sites. Numerical simulations have contributed significantly to our understanding of the behaviour of patchy particles, but, although simple in principle, advanced simulation techniques are often required to sample the low temperatures and long time scales associated with their self-assembly behaviour. In this work we review the most popular simulation techniques that have been used to study patchy particles, with a special focus on Monte Carlo methods. We cover many of the tools required to simulate patchy systems, from interaction potentials to biased moves, cluster moves, and free energy methods. The review is complemented by an educationally-oriented Monte Carlo computer code that implements all the techniques described in the text to simulate a well-known tetrahedral patchy particle model.

I. INTRODUCTION

The expression “patchy particle” has become more and more inclusive over the course of the years [1]. The most basic definition, “a colloid with attractive spots decorating its surface” is now outdated as patchy particles, or concepts developed in the field of patchy particles, are being used to model proteins [2–4], viral capsids [5], hard faceted bodies [6], double-stranded DNA [7] and even atoms and molecules [8, 9].

On the experimental side, tremendous progress is currently being made in the realization of such systems [10–15], and patchy particles have already been adopted by theorists as an ideal model to study the self-assembly properties of a variety of soft-matter systems [1, 16–19]. The number of examples of the impressive agreement between theory and simulations is rapidly increasing [20–24], establishing patchy particles as one of the most active and successful ideas in nanotechnology.

Molecular simulations played an important role in the development of the field, providing early predictions for many interesting new phenomena [25–30], which sparked the interest of the experimental community [10, 31–34]. Since the early days, many of the numerical and theoretical predictions, such as the shrinking of the gas-liquid phase separation region [22, 27], the existence of reentrant gas-liquid phase separations and reentrant gels [23, 35, 36], or the lack of crystallisation in highly

flexible patchy systems [37–39], have been observed in experiment. Even though it is becoming more and more common to consider patchy particles of increasing complexity [1], the majority of the theoretical breakthroughs have been obtained with toy models. Such simple models usually feature hard spherical particles and short-range patch-patch attractions. Notwithstanding the raise in popularity of systems composed of or inspired by patchy models, to the best of our knowledge a comprehensive resource to help choose and implement the right state-of-the-art algorithm to tackle the investigation of these systems is still lacking. This Review aims at filling such a gap.

In the following we give an overview of the more commonly used models and simulation methods that have been developed or adapted to patchy particles. We put special emphasis on Monte Carlo (MC) simulations, but we also mention models and algorithms pertaining to molecular dynamics (MD) simulations. In order to help the interested reader, we have implemented the most important MC algorithms described below in an open-source code, freely accessible on the web (<http://dx.doi.org/10.5281/zenodo.1153959>) [40]. This *PatchyParticles* code (from here on referred to as the *PP code*) has been developed with the idea of providing a simple and clear implementation rather than a fast and optimised one, and hence should be considered as an educational tool that complements the Review rather than a production-ready code.

* lorenzo.rovigatti@gmail.com

† john.russo@bristol.ac.uk

‡ flavio.romano@unive.it

II. PATCH-PATCH INTERACTION POTENTIALS

The inter-particle interaction in patchy systems is anisotropic and short-ranged. This fundamental aspect, which may or may not be complemented by a non-spherical shape [6, 7], is at the core of the complex behaviour of these systems. Here we review some of the most common potentials used to simulate toy models of patchy particles.

In what follows we will consider systems composed of spherical particles of diameter σ . Generalisations to systems that are polydisperse in size are straightforward, although uncommon [35]. By contrast, even though it is typical to simulate systems composed of identical particles, sometimes binary mixtures or even variability in the patch position, size or strength need to be taken into account [42–44]. Therefore, for the sake of generality we consider the surface of each particle i to be decorated with M_i patches. The position of patch α relative to the centre of mass of particle i , \mathbf{r}_i , is specified by the vector $\mathbf{r}_{\alpha,i}$. Let $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ be the distance between particle i and particle j and $\mathbf{r}_{ij,\alpha\beta} = \mathbf{r}_{ij} + \mathbf{r}_{\beta,j} - \mathbf{r}_{\alpha,i}$ be the distance between patch α on i and patch β on j . Furthermore, let $\{\mathbf{r}_{\alpha,i}\}$ be the set of vectors identifying all M_i patches of particle i . It is common to consider patchy particles that interact through an isotropic repulsive potential, that accounts for the mutually excluded volume, plus an attractive patch-patch term that depends on the relative orientations. By using the notation introduced above, the total interaction energy between particles i and j can thus be written as

$$V(\mathbf{r}_i, \mathbf{r}_j, \{\mathbf{r}_{\alpha,i}\}, \{\mathbf{r}_{\beta,j}\}) = V_{\text{ex}}(r_{ij}) + V_{\text{patch}}(\mathbf{r}_{ij}, \{\mathbf{r}_{\alpha,i}\}, \{\mathbf{r}_{\beta,j}\}) \quad (1)$$

where non-bold symbols indicate the moduli of the respective vectors. The excluded-volume term is, most of the times, a hard-sphere interaction for MC simulations or a differentiable hard-sphere-like potential (*e.g.* Weeks-Chandler-Anderson, inverse power) for MD simulations. However, sometimes screened electrostatic repulsions, modelled through Yukawa-like interactions, are also considered [18, 45].

The V_{patch} term is a sum of the contributions of all pair of patches, viz:

$$V_{\text{patch}} = \sum_{\alpha=1}^{M_i} \sum_{\beta=1}^{M_j} V_{\text{pp}}(\mathbf{r}_{ij}, \mathbf{r}_{\alpha,i}, \mathbf{r}_{\beta,j}). \quad (2)$$

It is common to employ attractive potentials that are, or resemble, square-well potentials (discontinuous or continuous, depending on the chosen simulation technique) and hence are functions of the patch-patch separation only, *e.g.* $V_{\text{pp}} = V_{\text{pp}}(r_{ij,\alpha\beta})$. This family of potentials, usually called sticky-spot or point-patch potentials, have essentially two parameters: the range of the interaction δ

and its strength, ϵ . An example of a continuous square-well-like attraction is [46]

$$V_{\text{pp}}(r_{ij,\alpha\beta}) = -\epsilon_{\alpha\beta} \exp \left[-\frac{1}{2} \left(\frac{r_{ij,\alpha\beta}}{\delta_{\alpha\beta}} \right)^{10} \right]. \quad (3)$$

A simple potential like the one of Eq. (3) contains the basic physics of patchy interactions: it confers a strong anisotropy and a well-defined valence to the particles. It has been used to look at the thermodynamics and dynamics of self-assembled disordered phases [18, 46–48]. However, as shown in Figure 1(a), patch-patch radial potentials have a spherical shape whose size is fixed by the single parameter δ , which therefore fully controls the bonding volume (or extent) of the patches. Consequently, the radial and angular flexibilities of a bonded pair cannot be tuned independently. As it turns out, these two quantities contribute differently to the entropy of disordered and ordered phases, going as far as determining their relative thermodynamic stability [37, 49]. Hence, in some cases it is important to be able to separately control the angular and radial flexibilities.

A well-known potential that provides control on the patch shape is the so-called Kern-Frenkel (KF) interaction [25]. It features two independent geometrical parameters, δ and θ^{max} , that set the radial and angular width, respectively. In the KF model patches can be visualized as spherical cones (see Figure 1(b)), with the patch-patch interaction being given by

$$V_{\text{pp}}(\mathbf{r}_{ij}, \hat{\mathbf{r}}_{\alpha,i}, \hat{\mathbf{r}}_{\beta,j}) = V_{\text{sw}}(r_{ij}) f(\mathbf{r}_{ij}, \hat{\mathbf{r}}_{\alpha,i}, \hat{\mathbf{r}}_{\beta,j}) \quad (4)$$

where V_{sw} is an isotropic square-well term of range $\sigma + \delta_{\alpha\beta}$ and depth $\epsilon_{\alpha\beta}$, the hat symbol $\hat{\cdot}$ indicate unit vectors and f is the orientation-dependent modulation term that takes the form

$$f(\mathbf{r}_{ij}, \hat{\mathbf{r}}_{\alpha,i}, \hat{\mathbf{r}}_{\beta,j}) = \begin{cases} 1 & \text{if } \hat{\mathbf{r}}_{ij} \cdot \hat{\mathbf{r}}_{\alpha,i} > \cos \theta_{\alpha\beta}^{\text{max}} \\ & \hat{\mathbf{r}}_{ji} \cdot \hat{\mathbf{r}}_{\beta,j} > \cos \theta_{\alpha\beta}^{\text{max}} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

During the course of the years, this potential has been used to investigate the phase behaviour of many types and blends of patchy and patchy-like systems [25, 26, 37, 43, 45].

We must note that having two independent parameters that model the extent of a patch-patch interaction arises naturally when dealing with hydrophobic surfaces, and indeed the KF model has been used to reproduce quantitatively the phase diagram and crystallization process of a quasi-2D system of hard particles with hydrophobic patches [50]. The reason is that two very short-ranged hydrophobic curved surfaces always have one contact point, which provides some negative energy. Where this contact happens on the two surfaces does not matter, and the KF model describes this kind of interaction.

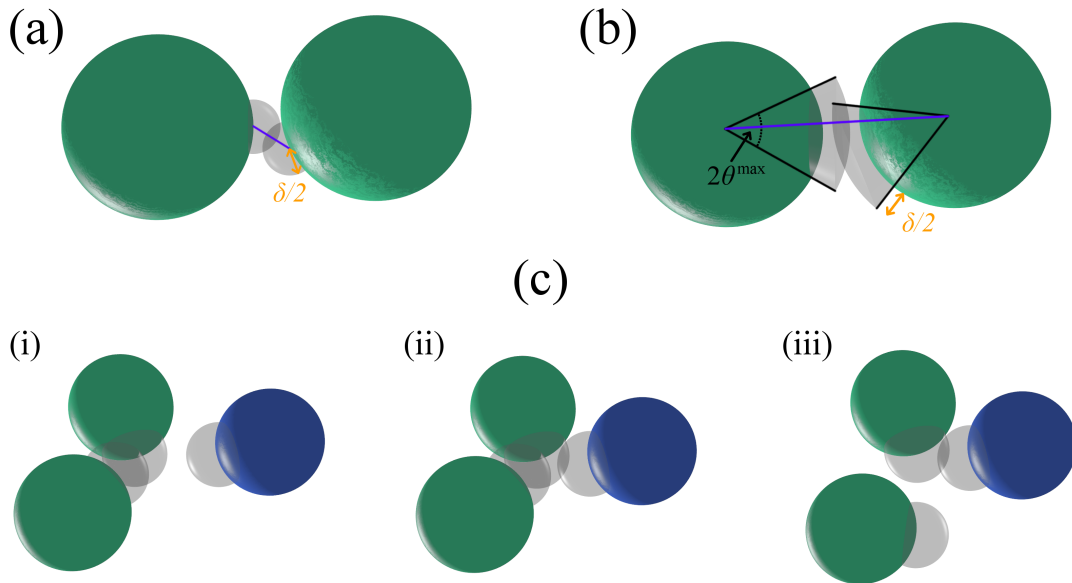


FIG. 1. A cartoon showing particles decorated with single patches, depicted in light grey. (a) In the sticky-spot model two patches interact only if their mutual distance (in violet) is smaller than two times the patch radius, $\delta/2$ (in orange). (b) In the Kern-Frenkel model patches are modelled as spherical cones of angular width $2\theta^{\max}$ and length $(\sigma + \delta)/2$, where σ is the particle diameter. Any two patches interact if they intersect with each other and if the vector connecting the centres of mass of the two particles (in violet) pass through both of them. (c) Thanks to the bond-swapping algorithm (see text and Ref. [41]), a free patch (here attached to the blue particle) can establish a bond with an already bonded patch whose original partner is free to move away at the end of the process. The attractive contribution due to the additional patch-patch bond present in (ii) is (partially or totally) counterbalanced by a three-body repulsion. Indeed, the energetic cost of the bond-swapping process, that is, its activation energy, is controlled by a parameter of the model and can be set to 0 make the reaction temperature-independent.

On the other hand, the point-patch model might seem more natural when dealing with particles that can experience a binding interaction depending on the position of the centre of the patch: one example is if the patches are realised through a single DNA strand that has some maximum length that it can reach given its attachment point. A key difference in between the two models, that must be taken into account when considering the binding properties of two particles is the fact that point-like models do not allow to freely rotate a particle around the vectors joining two bound particles, while the KF model does.

In order to exploit the advantages of the KF potential in MD simulations, continuous versions of the KF interaction have been developed. These are built by taking a differentiable isotropic (often square-well-like) function and by modulating it with functions that, akin to $f(\mathbf{r}_{ij}, \hat{\mathbf{r}}_{\alpha,i}, \hat{\mathbf{r}}_{\beta,j})$ in Eq. (5), depend on the scalar product between the particle-particle distance and the unit vectors that identify each patch. These functions are often Gaussians or generalised Gaussians, similar to the one of Eq. (3). These continuous potentials have been used to look at self-assembled ordered structures in patchy systems [29, 51–54], but they have been employed also in modelling patchy polymers [55] and even amino acids in proteins [2].

We note on passing that some of the behaviour dis-

played by patchy systems is also shared by other models that, in one way or another, fix the maximum number of bonds that each particle can form (the so-called *valence*). For instance, common potentials developed for water and silicon employ rigid bodies complemented by charges or spheres interacting through three-body interactions that limit the ideal number of favourably interacting neighbours [56–58].

III. THE SINGLE-BOND-PER-PATCH CONDITION

The single-bond-per-patch condition (SBPPC) is essential to control the valence. The valence, in turn, is the most important property in determining the phase behaviour of patchy particles. Most models enforce the SBPPC through a judicious choice of the model geometry and parameters. For rather simple, and thus computationally efficient, models (such as spheres decorated with patches) the SBPPC requires quite small patches, and hence very small bonding volumes. For instance, the single geometrical parameter of the point-patch potential, the patch-patch attraction range, has to be smaller than 0.119σ (σ being the particle diameter) to enforce the SBPPC. By contrast, the two parameters of the Kern-Frenkel potential, the range δ and

the angular width θ_{\max} , have to fulfil the inequality $\sin(\theta_{\max}) \leq (2(1 + \delta\sigma))^{-1}$. Under these constraints, the largest bonding volumes attainable are of the order of $\approx 10^{-3}\sigma^3$. Such small values are problematic from a technical and a fundamental point of view. Indeed, from a computational standpoint small bonding volumes make it very hard to equilibrate at low T . From a modelling perspective, instead, some very interesting phenomena (such as the appearance of disordered ground states [37] or of reentrant gas-liquid phase separations [59]) require or are accessible with large bonding volumes only. Furthermore, some “soft” building blocks, such as telechelic star polymers or DNA nanostars, possess an intrinsically large bonding volume due to their inner flexibility [1]. The question that arises is whether it is possible to take into account such a high degree of internal flexibility in a straightforward way by devising models that are nearly as simple as the ones described above.

The issue regarding the precise value of the bonding volume can be worked around in Monte Carlo or event-driven simulations, where non-continuous constraints are easy to implement, by putting an explicit limit to the number of bonds per particle [60, 61]. Operatively, this is done by adding an attractive term to the interaction between two particles only if both have fewer than the maximum number of allowed bonds. A drawback of this approach is that the resulting model can be hardly seen as realistic and, importantly, is not amenable to be investigated *via* MD simulations.

A more sophisticated approach has been introduced in Ref. [41] and employed to readily generate fully-bonded disordered configurations to model microgel particles [62], and to investigate the dynamics of a simple model of vitrimers at large length- and time-scales [63]. The idea behind the method is to add a short-ranged three-body repulsive term to triplets of close patches in order to compensate the additional (negative) energy contribution due to the formation of extra bonds. The approach is compatible with continuous potentials, meaning that it can be used in molecular dynamics simulations. The model has a parameter, λ , that controls the energetic penalty that non-bonded patches have to pay in order to get close to a bonded pair. The model thus not only enforces the SBPPC but also introduces a bond-swapping mechanism that facilitates the dynamics at any T .

A cartoon depicting a bond-swapping process is presented in Figure 1(c). Taking three patches A , B , and C as an example, if $\lambda = 1$, then the energy of configurations where A is bonded to both B and C (panel (ii)) or just to one of them (panels (i) and (iii)) is the same. Under this condition, the process through which the $A - B$ bond breaks and the $A - C$ bond forms is carried out at constant energy: temperature plays no role and the system can rearrange its bonding pattern even in the limit $T \rightarrow 0$. By contrast, if $\lambda > 1$, the bond swapping process becomes energetically expensive and hence thermally activated.

IV. MONTE CARLO MOVES

Rototranslations and particle insertions/deletions, which are briefly introduced in the paragraphs below, are the simplest moves that allow to sample the canonical and grand canonical ensembles, respectively, for any system. However, self-assembly processes in patchy systems often occur when the bonding energy ϵ is substantially larger than the thermal energy $k_B T$ (see *e.g.* Refs. [23, 26, 29]). For instance, it is common to simulate patchy systems with $\epsilon/k_B T \gtrsim 5$. Under these conditions, most of the patches are involved in bonds, and the Metropolis acceptance of unbiased rototranslations that attempt to break a single bond, $\exp(-\beta\Delta E)$, plummets from $\approx 7 \times 10^{-3}$ down to $\approx 4.5 \times 10^{-5}$ as the bonding strength relative to the thermal energy increases from $\epsilon/k_B T = 5$ to $\epsilon/k_B T = 10$. As a result, the great majority of the moves will very likely be rejected, thereby greatly reducing the efficiency of the simulation. This issue can be overcome, or at least improved upon, by employing *biased* Monte Carlo moves [64]. After a brief introduction of the most basic MC moves, we will discuss two biased MC moves that are not system-specific and can greatly enhance the effective exploration of the phase space.

A. Rototranslations

When dealing with anisotropic particles, a good sampling of the phase space requires a correct handling of both rotations and translations. In Monte Carlo simulations, the simplest moves that ensure equilibrium are pure translations and pure rotations or, as done in the PP code, rototranslations, which are combinations of the two. For both translations and rotations it is common to set the maximum (radial or angular) displacement so as to have an acceptance ratio of about 0.2 – 0.4, with the optimal value being dependent on the specific system investigated [64, 65]. Note that, in force of the short-range nature of the interaction potentials (see Section II), these maximum displacements will necessarily be small.

A rototranslation is carried out by choosing a random particle i , random angular and radial displacements and a random axis of rotation. Particle i is then moved and rotated according to the values extracted. The difference in energy due to the move is then used to compute the Boltzmann factor which, in turn, is connected to the acceptance probability, as per the standard Metropolis Monte Carlo [64]. The only choice that needs to be made is how to handle the rotational degrees of freedom. Quaternions and explicit orientation matrices are both common choices [66, 67]. Note that floating-point arithmetic tends to disrupt the orthonormality of the data structures that store the orientational degrees of freedom. Care has to be taken to ensure that the quality of the orthonormality does not degrade too much over the course of the simulation. The `MC_move_rototranslate` function

of the `MC.c` file shows how rototranslations are implemented in the PP code, which employs rotation matrices that are re-orthonormalised through a Gram-Schmidt procedure every time that the energy is printed (see the `MC.check_energy` function).

Compared to MC, rotations are a more delicate matter in MD simulations. Indeed, when dealing with rigid bodies one has to consider that particle-particle interactions exert not only central forces but also torques. In many cases these torques have complicated expressions. However, there exist guidelines that, depending on the specific type of interaction, can greatly simplify the deriving procedure [68]. The algorithm used to integrate the equations of motion can be also carried out in different fashions, depending on whether quaternions, rotation matrices or constraint algorithms are employed [66, 69–72] and on the type of dynamics [73–75]. A drawback of the most common patchy interactions (see Section II) is their steepness, which negatively affects numerical stability and makes it hard (or even impossible) to use single-precision floating point arithmetic (which some GPU-powered packages offer to improve performance [76, 77]). For the same reason, extra care has to be taken when choosing the integration time step.

B. Grand Canonical ensemble

In the grand canonical ensemble (GCE) the system is in equilibrium not only with a thermal bath but also with a reservoir of particles [64]. Therefore, the three thermodynamic quantities that are kept fixed are volume V , temperature T and chemical potential μ . As a result, the overall number of particles N fluctuates. The GCE is commonly used in Monte Carlo simulations to investigate the phase behaviour of disordered systems [78], and can be readily extended to many-component mixtures [79, 80]. In addition, GCE simulations can be combined with more sophisticated techniques, such as successive umbrella sampling (see Section VB) or Wang-Landau sampling [81] to overcome free-energy barriers.

In GC simulations there are two additional trial moves that attempt to either add or remove a particle. Note that for the particle addition move, the new particle must be added in a random position with a randomly distributed orientation. The two MC acceptance probabilities read

$$\text{acc}(N \rightarrow N+1) = \min \left(1, \frac{Vz}{\Lambda^3(N+1)} e^{-\beta \Delta E} \right) \quad (6)$$

$$\text{acc}(N \rightarrow N-1) = \min \left(1, \frac{\Lambda^3 N}{Vz} e^{-\beta \Delta E} \right) \quad (7)$$

where Λ is the thermal wavelength, $\beta = 1/k_B T$, $z = e^{\beta \mu}$ is the activity and $\Delta E = E_{\text{new}} - E_{\text{old}}$ is the energy difference between the final and initial configurations. The implementation of the main core of the GCMC algorithm

can be found in the PP code in the `MC.add_remove` function of the `MC.c` file.

C. Aggregation-Volume-Bias moves

Self assembly processes often require low density and temperature, specially in patchy systems. The sampling of the phase space under these conditions can be dramatically improved with the so-called aggregation-volume-bias (AVB) Monte Carlo moves [82, 83].

The AVB scheme provides two different basic moves. Detailed balance requires that both types of moves are used during the course of the simulation. The first one (hereafter referred to as the AVB-B move) attempts to form a bond between two previously unbonded particles, while the second one (the AVB-U move) attempts to break an existing bond by separating a bonded pair. Here the notion of two particles being “bonded” or “unbonded” should fulfil a criterion that need not be necessarily related to the specific model employed. However, it is often convenient to use an operative definition for the bonding between two particles that coincides with a state of low pair energy. For instance, an optimal criterion for the KF interaction is to regard two particles as bonded when they share a patch-patch bond. Once the bonding criterion has been set, the bonding (phase-space) volume is defined as the number of microscopic configurations for which two particles are bonded, V_{AVB} . According to this definition, V_{AVB} depends only on the bonding criterion and not on the macroscopic thermodynamic variables (such as T , μ , N , V , *etc.*). For instance, the expressions for the bonding volume of a KF $\alpha\beta$ bond in 3D and 2D are:

$$V_{\text{AVB}} = \frac{\pi \sigma^3}{3} (1 - \cos \theta_{\alpha\beta}^{\text{max}})^2 [(\sigma + \delta_{\alpha\beta})^3 - \sigma^3] \quad (3D)$$

$$V_{\text{AVB}} = \frac{(\theta_{\alpha\beta}^{\text{max}})^2}{\pi} [(\sigma + \delta_{\alpha\beta})^2 - \sigma^2] \quad (2D).$$

We then define the “outer” (phase-space) volume V_O as the number of configurations for which two particles are not bonded. It follows from its definition that $V_O = 4\pi V - V_{\text{AVB}}$, where V is the volume of the simulation box and 4π comes from the rotational degrees of freedom. These two volumes will be used in the following to bias the acceptance of the two moves.

Finally, let N_i be the number of particles that are bonded to particle i . The recipe for the AVB-B move is then:

1. Randomly select a particle i .
2. Randomly select a particle j which is neither i nor one of its bonded neighbours.
3. Move particle j inside the bonding volume of particle i .

4. Accept the move with probability

$$\min \left(1, \frac{(N - N_i - 1)V_{\text{AVB}}}{(N_i + 1)V_O} \exp(-\beta\Delta E) \right).$$

Note that the insertion of particle j in the bonding volume of particle i must be carried out uniformly. In other words, the probability that, at the end of a trial move, i and j are in a specific mutual arrangement must be the same for every microscopic configuration that fulfils the bonding criterion. The PP code shows how to perform this operation for KF patches in the `place_inside_vbonding` function of the `utils.c` file.

In order to perform an AVB-U move one has to

1. Randomly select a particle i .
2. Randomly select a particle j that is bonded with i . If i has no neighbours (that is, if $N_i = 0$) then outright reject the move.
3. Move particle j outside of the bonding volume of i , so that at the end of this step i and j are no longer bonded.
4. Accept the move with probability

$$\min \left(1, \frac{N_i V_O}{(N - N_i) V_{\text{AVB}}} \exp(-\beta\Delta E) \right).$$

Note that step 3 can be performed by randomly inserting j in the box with a random orientation till i and j are no longer bonded.

The ratio between the two volumes defined above, $r \equiv V_{\text{AVB}}/V_O$, is what biases the acceptance of the AVB-B move. Since the particle bonding volume is always (much) smaller than the overall volume, the bias *lowers* the acceptance probability of the move. However, at low temperature this is rendered unimportant by the very large value of the Boltzmann factors associated to the creation of additional bonds. By contrast, $1/r$, which is much larger than 1, biases the acceptance of the AVB-U move. Therefore, the larger the value of r , the better AVB will perform compared to a standard MC algorithm. As a result, the AVB algorithm shines when used in conjunction with short-ranged potentials (small V_{AVB}) and low-density systems (large V_O). There is an additional reason why AVB moves perform better at low density: it dramatically increases the speed with which particles move around the simulation volume, which is a common efficiency bottleneck for low-density systems when only single-particle rototranslations are employed.

We note on passing that there exists a third type of AVB moves which takes a particle i bonded with particle j and insert it in the bonding volume of a third particle [83]. It can greatly enhance the sampling in specific cases such as chain-forming systems [59].

In order to benchmark the performance of the AVB scheme we simulate several systems made of $N = 500$

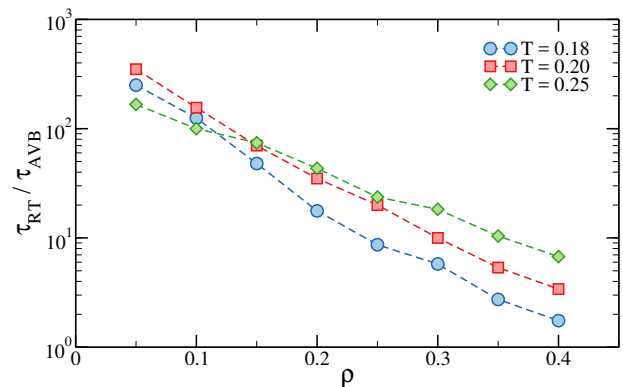


FIG. 2. The ratio between the autocorrelation time of the energy of systems simulated without and with AVB moves as a function of density for different T . Assuming that the computational cost of the two procedures is the same (see text), this ratio represents the gain achieved when using AVB.

tetravalent patchy particles at different ρ and T . For each one we compute the autocorrelation function of the energy. We then estimate the time τ at which the autocorrelation function takes the value $1/e$ and use it as a proxy for the simulation efficiency. Figure 2 shows the ratio between the autocorrelation time for systems where AVB moves were not attempted, τ_{RT} , and attempted 50% of the time, τ_{AVB} . As mentioned above, the AVB scheme performs better at low density. For very small values of the density ($\rho \leq 0.1$), which are nonetheless still relevant to many self-assembly processes [46], AVB enhances the efficiency of the simulation by more than two orders of magnitude. By contrast, the dependence on temperature is less dramatic, albeit non-monotonic, and more system-dependent.

A basic implementation of the AVB algorithm is contained in the `avb.c` and `avb.h` files of the PP code. We note that the PP code runs roughly 20% – 40% slower when AVB moves are enabled. This is merely an implementation issue and mostly due to the lack of optimisation that stems from the educational nature of the PP code. With a properly optimised code the AVB scheme is only marginally slower than a regular Monte Carlo simulation.

We conclude this section by mentioning that AVB moves and GCMC simulations complement each other very well. Indeed, when simulating low-temperature systems the most probable GC deletions are those attempted on monomers. Unfortunately, under normal conditions it is most likely to attempt to remove a monomer generated by a previous GC addition rather than a monomer that spontaneously broke off all of its bonds. As a result, many GC moves are essentially wasted. However, when the AVB scheme is applied the rate of monomer formation/depletion is greatly enhanced by the AVB-U and AVB-B moves, respectively, and thus the number of effective GC moves increases.

D. Virtual Move Monte Carlo

The Virtual Move Monte Carlo (VMMC) is a Monte Carlo cluster-move algorithm proposed originally by Whitelam and Geissler [84] and designed to improve relaxation times in strongly-interacting, low-density systems and to better approximate diffusive dynamics in these systems. As it often happens, the algorithm has proved to be useful in many other contexts, such as polymeric systems [85, 86]. The VMMC cluster move, quite complicated to implement in its original version, has been presented in a more simple way by the same authors [87], and more recently an easier version to program has been published by Růžička and Allen [88]. The idea behind the algorithm is to avoid one of the shortcomings of standard cluster algorithms, whereby the clusters are built based upon the system microstate *before* the move, and any move that attempts to merge or break clusters has to be rejected in order to satisfy detailed balance. This means that cluster moves need to be mixed with standard single-particle moves, and although they do speed up the diffusion of clusters, the pattern of the interactions is changed mostly by single-particle moves. In the context of patchy particles, where patch-patch bonds are the natural criterion to define clusters, cluster moves based only on the system microstate before the move will never change the bonding pattern by construction. To overcome this limitation, Whitelam and Geissler have designed the VMMC cluster move, a “virtual move” that builds a cluster based upon the *move* as well as the starting microstate: particles are recruited into the cluster depending on how much their energy changes with the move, rather than how strongly they interact with their neighbours. Essentially, the cluster is chosen on the fly by starting to move a particle and then recruiting all those neighbouring particles that would rather move than remain still and pay an energetic cost.

Here we describe the most recent iteration of the algorithm proposed in Ref. [88], implemented in the function `VMMC_dynamics` of the `VMMC.c` file of the PP code.

1. Choose a move (a random displacement for translations, an axis and an angle for rotations).
2. Choose a seed particle for the move, and add it to the cluster. The cluster will contain all those particles that will be moved.
3. Build or update a list \mathcal{L} of particle pairs that will have their pair energy changed if the cluster is moved. Note that, by construction, one and only one particle in each pair belongs to the cluster, since the cluster moves as a single object and hence the pair interaction energy of particles that belong to the cluster will not be affected by the move. Also note that \mathcal{L} will contain particle pairs that are affected from the cluster moving both away and towards them.

4. Choose a random pair $l = (i, j)$ in \mathcal{L} . Call i the particle in the pair l that belongs to the cluster and j the particle that does not. Remove this pair from the list.
5. Compute (or recover from bookkeeping) $u(i, j)$, the energy of the pair in the starting microstate.
6. Compute $u(i', j)$, the energy of the pair after i has been moved and j has not.
7. Compute $u(i, j')$, the energy of the pair after j has been moved and i has not.
8. Extract a random number $r \in [0, 1]$. If $\exp\{-\beta(u(i', j) - u(i, j))\} > r$, insert j in a list \mathcal{C} of particles that are candidates for recruitment in the cluster.
9. Extract a new random number $q \in [0, 1]$. If $\exp\{-\beta(u(i, j') - u(i, j))\} > q$, recruit j into the cluster and remove it from \mathcal{C} .
10. If \mathcal{L} is not empty, go to step 3. If \mathcal{L} is empty, the cluster has been built. Please note that the cluster may contain just the seed particle.
11. If \mathcal{C} is not empty, reject the move. This step is necessary to enforce balance [84]. Otherwise, accept the cluster move.

We point the reader to Ref. [84] and Ref. [88] for a detailed discussion of how this algorithm is derived. Here we will just elaborate on some of the less obvious points of the algorithm. For example, the random choice of an element in \mathcal{L} , rather than processing the pairs as the algorithm finds them, is needed to ensure super-detailed balance, *i.e.*, that the algorithm is in principle able to perform the exact reverse move immediately. Also, the fact that \mathcal{C} needs to be empty for the move to be accepted is done for the same reason. The quantities in steps 8 and 9 are chosen in such a way to build a cluster move that will be automatically accepted in step 11 if \mathcal{C} is empty.

Although significantly more difficult to implement than a standard single-particle or cluster move, this algorithm has a number of benefits that make its use very beneficial in a variety of situations. The fact that it is the move that dictates the cluster allows for internal relaxation of strongly-interacting regions: particles are recruited into the cluster only if not doing so would increase the energy (on average). Strongly interacting pairs are not necessarily recruited together: in the extreme case of $u(i, j)$ very negative, if $u(i', j)$ is also very negative (the relative move does not affect the energy) particle j is *not* recruited into the cluster (see step 8). This allows, for example, for relative rotation of clusters or for naturally arising pivot and crankshaft moves in polymeric systems.

In our experience, we have found that VMMC is a good algorithm for self-assembling and polymeric systems due precisely to its ability of finding the natural moves that

help relaxing the system. On the other hand, the algorithm has a few drawbacks. Firstly, care must be taken in the choice of the move at point 1 above: a large move will easily recruit the whole system into the cluster, essentially wasting computer time by building a large cluster for a move that, even if accepted, is effectively useless. This would not happen frequently in the low-density scenario for which the algorithm was designed, but nevertheless it should be taken into account. Secondly, the algorithm is not straightforward to program in an efficient way.

In terms of performances, the original paper [84] highlights that the VMMC algorithm is best suited for low-density, low-temperature systems, where the system is largely composed of clusters of particles. In particular, VMMC greatly speeds up the equilibration of inter-cluster interactions essentially by promoting a much faster diffusion of clusters than single-particle MC moves. Importantly, clusters can merge and break while moving, which is forbidden in standard cluster moves. Since each virtual move is much more computationally expensive than a standard single-particle move, the benefits of VMMC are worth the extra effort only when the diffusion of clusters is the bottleneck of the simulation.

V. FREE-ENERGY METHODS

The level of control and tunability of patchy models has opened the way to a deeper understanding of the phase behaviour of complex systems with directional interactions [1, 89–91]. For the disordered phases (liquid and gas), patchy models were developed to study both theoretically and computationally the process of self-assembly, with particular emphasis on the polymerization [92] and gel transitions [46]. For ordered (crystal) phases, patchy particles have been shown to be able to describe the crystallization of open crystalline structures [20, 93, 94], most notably the diamond crystal for 4-patch models [29, 44, 49]. Self-assembly can have a deep impact on the phase behaviour of the system, and patchy particles were the ideal platform where new transitions and new types of phase diagrams were discovered. Empty liquids [27], re-entrant and topological phase transitions [59], amorphous ground states [37], are all examples where new types of phase behaviour were discovered in the context of patchy particles.

Most notably these discoveries were made possible by the use of highly-efficient free-energy methods. We should note that free-energy methods can be divided into two broad categories: i) the ones that attempt to recover the free energy of a thermodynamic phase, and ii) the ones that aim at finding the *potential of mean force*, i.e. the projection of the free energy along a *reaction coordinate* or *order parameter*. In the next sections we will briefly discuss the former and then describe more in depth one of the latter.

A. Direct evaluation

A phase diagram of a substance shows under which thermodynamic conditions a phase (or a set of phases) is in equilibrium. From a simulation standpoint, there exist several methods to straightforwardly estimate the phase boundaries between disordered phases (one of which will be thoroughly discussed in the next section) [95, 96]. By contrast, the presence of ordered phases greatly complicates the numerical evaluation of the phase behaviour. Quite remarkably, the main challenge is not posed by the actual calculation of the relative free energy between the phases (which, in turn, dictates their relative stability), but by the fact that, in general, it is not possible to know *a priori* what these phases are in terms of symmetry, unit cells, *etc.* Indeed, while for simple particles such as hard spheres an educated guess does often suffice, more complicated building blocks such as patchy or non-spherical particles might require approaches that go beyond “hand”-based or brute-force procedures. An efficient way of searching the vast space of possible lattice candidates is provided by the so-called floppy box method. The basic idea is to simulate a small number of particles in a box of variable shape in the isothermal-isobaric ensemble [97], and collect the most stable structures in the range of temperatures and pressures of interest. This method has been applied to, among others, non-convex objects [98, 99] and one-patch particles [100]. In parallel, the idea of leveraging genetic algorithms to efficiently compile a list of candidate structures has surfaced, and it has been employed in patchy systems to find those ordered arrangements that are most favourable from an energetic standpoint [101, 102]. The two methods tend to perform similarly, and the best choice depends on the system under investigation [103].

Once the most promising candidate structures have been singled out, their relative stability is evaluated by computing the free-energy difference with respect to a system of known free energy. This is usually done by employing a combination of Hamiltonian and thermodynamic integration techniques. Well-known methods that build upon this idea are the Frenkel-Ladd [104] and Einstein Molecule [105] methods. A more thorough discussion on this topic falls out of the scope of the present review, and we refer the interested reader to Ref. [106].

B. Potential of mean force

Another approach is to reconstruct the free-energy landscape by sampling rare fluctuations of one or more relevant order parameters. The last decade has seen a flourishing of such methods, as for example Wang-Landau sampling [95] and Metadynamics [107, 108]. Here we focus on methods based instead on the Umbrella Sampling technique, which have had the biggest impact on the literature on Patchy Particles. In this section we first briefly review the basics of the Umbrella Sampling

method, and then focus on a variant (Successive Umbrella Sampling) which has found widespread adoption.

Umbrella Sampling

Umbrella sampling (US) [109, 110] is a method for enhancing the sampling of regions of configurational space which have a small weight (as measured by the Boltzmann distribution) under equilibrium conditions. By enabling the sampling of rare fluctuations, Umbrella Sampling allows one to reconstruct the free-energy profile even in regions of low statistical weight, such as on top of free-energy barriers. This is achieved by modifying the Markov chain with an external potential term which depends on the relevant order parameter for the transition of interest. The most immediate advantage of US over its competitors is that simulations of a system with a biased potential run in equilibrium, and the bias can be removed to recover the unbiased (true) probability distribution of the order parameter.

In Umbrella Sampling, a biasing potential $\eta(x)$ is added to the Hamiltonian of the system

$$\mathcal{H}' = \mathcal{H} + \eta(x)$$

where \mathcal{H} is the base Hamiltonian (*e.g.* the patchy potential), and x is the order parameter whose distribution $\mathcal{P}(x)$ we wish to measure. By running a simulation with the new Hamiltonian \mathcal{H}' , and measuring its order parameter distribution, $\mathcal{P}'(x)$, we can readily recover the original distribution as

$$\mathcal{P}(x) = \mathcal{P}'(x) \exp(\beta\eta(x)) \quad (8)$$

By choosing an appropriate $\eta(x)$ it is thus possible to sample the order parameter for arbitrary values of x , and then recover the original unbiased distribution with Eq. (8). A standard choice is to use an harmonic potential, $\eta(x) = k(x - x_0)^2$, that forces the system to sample regions around $x = x_0$ with fluctuations that depend on the coefficient k . A typical use of Umbrella Sampling to estimate the free energy landscape is to perform several biased simulations centered around adjacent values of x_0 , such that the fluctuations of the order parameter x of two consecutive simulations have a significant overlap. The full distribution $\mathcal{P}(x)$ is then obtained by combining the order parameter histograms of different simulations together. A statistically self-consistent way to do this is *via* the Weighted Histogram Analysis method [111]. From knowledge of the distribution function $\mathcal{P}(x)$ we can then obtain the free energy (or potential of mean force) as

$$F(x) = -k_B T \log \mathcal{P}(x) \quad (9)$$

Umbrella Sampling is an equilibrium method that should give the correct free energy independently of the choice of the biasing potential. However, in practice, care

has to be taken to ensure appropriate sampling of the configurational space. When dividing the order parameter domain in several intervals (each centered at a different value of x_0), ensuring a proper sampling within each window often requires choosing different potentials for each window. In the case of harmonic potentials, for example, different intervals often require different values of the elastic constant k depending on the slope of the free energy around x_0 . The higher the slope of the free energy, the higher the value of k one needs to set in order to sample correctly the configurations around x_0 . But since the slope of the free-energy is not known *a priori*, one often has to tune the values of k during the course of the simulation. To minimize this problem, two diametrically opposite strategies have been adopted.

One possibility is to sample the free energy landscape in as few intervals as possible. A way to do this is to use specific forms of the biasing potential based on theoretical expectations for the barrier. We give here an example for crystal nucleation processes. A good order parameter for crystallization processes is the size of the largest crystalline cluster [112, 113], which we denote here as x . Starting from the metastable liquid phase ($x = 0$), we want to use Umbrella Sampling to access rare fluctuations, corresponding to the appearance of a crystalline nucleus of size x . Classical Nucleation Theory [114] predicts the free energy barrier for nucleation as

$$\beta\Delta F(x) = -|\Delta\mu|x^{2/3}(x^{1/3} - 1.5x_c^{1/3}) \quad (10)$$

where $\Delta\mu$ is the chemical potential difference between the two phases, x is the number of particles in the largest crystal nucleus, and x_c is the critical nucleus size. In order to explore states on top of the barrier, we modify the Hamiltonian with a biasing potential that exactly cancels the barrier

$$\eta(x) = -\beta\Delta F(x) \quad (11)$$

This variant, called Classical Nucleation Theory Umbrella Sampling scheme (CNT-US), has been introduced in Ref. [114]. The only parameter needed is an estimate of $\Delta\mu$. This can be obtained for example with the following relation $\Delta\mu = H_{\text{fusion}}(1 - T/T_m)$ (where H_{fusion} is the enthalpy of fusion, while T_m is the melting temperature), or by computing the absolute free energies of the bulk phases, as described in Section V A. Once a reasonable approximation of $\Delta\mu$ is obtained, several independent simulations at different values of x_c are run. When x_c is close to the true critical size, the corresponding simulations will exhibit large fluctuations, as the bias in Eq. 11 is effectively cancelling the nucleation barrier. All other simulations (which exhibit small fluctuations in x) are quickly discarded. Instead, the simulation which maximizes the fluctuations is used to reconstruct the whole barrier by using Eqs. 8 and 9.

The second approach to free energy calculations is embodied by a technique called Successive Umbrella Sampling (SUS). Here, instead of targeting a uniform sampling of a large portion of the order parameter space, the

aim is to divide the order parameter space in as many small intervals as possible.

Successive Umbrella Sampling

In Successive Umbrella Sampling [96] simulations, the order parameter space is divided in many small intervals that are sampled consecutively. While the approach is general, we will present it here for the choice $x = N$, i.e. when the order parameter is simply the number N of particles in the simulation. This choice is appropriate for those transitions in which the density $\rho = N/V$ is a good order parameter. To sample fluctuations in the number of particles N , each SUS simulation is performed in the grand-canonical ensemble. In the context of patchy particles, the method has been used to study liquid-gas transitions, for example in re-entrant fluids [59], mixtures [43], dipolar interactions [115], and isotropic-nematic phase boundaries [116].

In SUS, the order parameter N is divided in many intervals (also called windows) of size k , such that the number of particles allowed in each interval i is

$$N \in \{N_i, N_{i+1}, \dots, N_{i+k-1}\}$$

For each interval, a simulation is run where its number of particles is constrained by reflecting boundary conditions at N_{i-1} and N_{i+k} . In other words, grand-canonical moves that would bring the number of particles outside the allowed range are rejected. The size of each interval k is often set equal to $k = 2$ to maximize the number of intervals, and minimizing the overlap between them

$$\{1, 2\}, \{2, 3\}, \dots, \{N-2, N-1\}, \{N-1, N\}$$

During a simulation sampling the interval i , the number of times N was visited is used to construct the histogram $H_i(N)$. If a move is rejected due to reflective boundary conditions, the histogram is still updated for the value of N . In the simplest implementation, all windows are simulated independently and without the use of biasing potentials.

In each interval (assuming without loss of generality that $k = 2$) the free energy difference between N_i and N_{i+1} is given by

$$F(N_{i+1}) - F(N_i) = -k_B T \log \frac{H_i(N_{i+1})}{H_i(N_i)}$$

The total free energy is obtained by combining the different intervals

$$F(N) - F(0) = -k_B T \log \frac{\mathcal{P}(N)}{\mathcal{P}(0)}$$

where $\mathcal{P}(N)$ is the unnormalised probability distribution

$$\frac{\mathcal{P}(N)}{\mathcal{P}(0)} = \frac{H_N(N)}{H_N(N-1)} \frac{H_{N-1}(N-1)}{H_{N-1}(N-2)} \dots \frac{H_1(1)}{H_1(0)}. \quad (12)$$

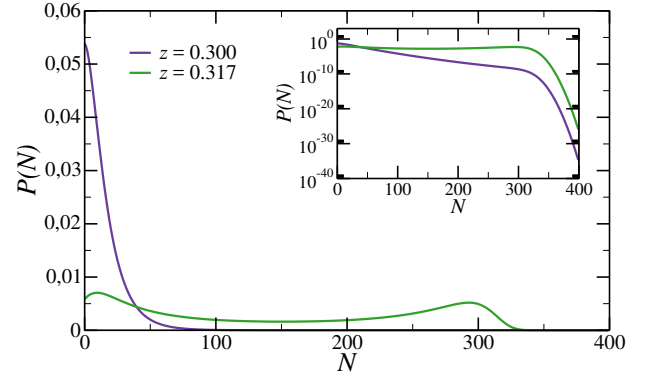


FIG. 3. The probability distribution of having N particles, $\mathcal{P}(N)$, for a system of KF tetravalent patchy particles with $\delta = 0.26$ and $\cos \theta^{\max} = 0.92$ simulated at $z = 0.300$ and $T = 0.160$, below the critical temperature (violet curve). Here we also show the $\mathcal{P}(N)$ reweighted so that the area below the peaks of the two coexisting phases is equal (green curve). The inset shows the same data on a log-lin scale to show the tens of orders of magnitude accessed.

The SUS technique provides some definitive advantages over traditional Umbrella Sampling simulations. The first one is that, since the windows are very small, even unbiased simulations are able to sample all the states within the window, without the use of auxiliary potentials. If needed, umbrella potentials can still be used, and it is often possible to extrapolate the potential in one window from the measured free-energy difference in the previous window. The second big advantage is that SUS allows to trivially (and massively) parallelize the computation of free energies, as the calculation is divided in a large number of independent windows. All this while retaining the advantage of the traditional Umbrella Sampling scheme, which is the use of equilibrium simulations that can run without any prior knowledge of the free-energy barrier.

An example of the output of a SUS simulation performed with the PP code is shown in Fig. 3. Here the system under study is composed of tetravalent patchy particles interacting through a KF potential with parameters $\delta = 0.26$ and $\cos \theta^{\max} = 0.92$, simulated at $T = 0.160$, $V = 667 \sigma^3$ and $z = 0.3$. Since the simulation has been run at a sub-critical temperature, there exist some values of z for which the $\mathcal{P}(N)$ is double-peaked. In general, SUS simulations allow for an accurate sampling over many orders of magnitude (see inset of Figure 3), which makes it possible to evaluate the probability distribution at different z by histogram reweighting techniques [111]. Here we have changed z until the area below the peaks associated to the gas and liquid phases is equal (see green curve in Fig. 3), a condition which is commonly associated to phase coexistence.

In the PP code, the histogram is updated in the `do_SUS` function of the `MC.c` file, while the GC moves, and the logic for the rejection of the moves that attempt to break out of the window's boundaries, are implemented in the

function `MC_add_remove` of the same file.

VI. CONCLUSIONS

In this paper we have reviewed some of the most common models and most useful techniques that have been used to simulate systems of patchy particles, with a special focus on Monte Carlo methods. In contrast with atomic and molecular systems, when simulating patchy particles the computationally intensive part is not the evaluation of the pair interactions or of the forces, since the potential is usually simple and very short ranged. On the other hand, the short-range nature of the interaction sets the smallest length scale that needs to be sampled, forcing very small single-particle moves in Monte Carlo simulations and small values of the time integration steps in molecular dynamics.

A further challenge resides in the fact that the interesting behaviour in this class of systems is almost inevitably found at very low temperature, *i.e.*, when a bond between two particles is of the order of 5-10 times (or more) the thermal energy. This contributes to make it challenging to achieve equilibrium without the use of advanced

Monte Carlo techniques such as the ones described here.

Since it is impossible to devise a Monte Carlo move that will speed up sampling for any system and state point, we have focussed on techniques that are general enough to be useful in many use cases and showed how they perform on a typical and well known patchy particle system. Since some of the techniques treated are not straightforward to implement, we also provide an (as simple as possible) implementation in a tutorial code freely available online [40]. We believe such educational tool to be a useful starting point to build up bespoke codes tailored to study systems that would not be treatable otherwise.

ACKNOWLEDGEMENTS

All authors are greatly indebted to Francesco Sciortino, whose vision and brilliance are a constant source of inspiration. LR acknowledges support from the ERC Consolidator Grant 681597 MIMIC. JR acknowledges support from the ERC Grant DLV-759187 and the Royal Society University Research Fellowship. We thank V. Bianco, P. Handle and E. Locatelli for their insightful feedback.

-
- [1] E. Bianchi, B. Capone, I. Coluzza, L. Rovigatti, and P. D. J. van Oostrum, *Phys. Chem. Chem. Phys.* **19**, 19847 (2017).
 - [2] I. Coluzza, *PLoS ONE* **9**, e112852 (2014), arXiv:1406.4373v1.
 - [3] J. J. McManus, P. Charbonneau, E. Zaccarelli, and N. Asherie, *Curr. Opin. Colloid Interface Sci.* **22**, 73 (2016).
 - [4] J. Cai, J. P. Townsend, T. C. Dodson, P. A. Heiney, and A. M. Sweeney, *Science* **357**, 564 (2017).
 - [5] D. C. Rapaport, *Phys. Rev. Lett.* **101**, 186101 (2008).
 - [6] J. A. Millan, D. Ortiz, and S. C. Glotzer, *Soft Matter* **11**, 1386 (2015).
 - [7] C. De Michele, T. Bellini, and F. Sciortino, *Macromolecules* **45**, 1090 (2012).
 - [8] S. Whitelam, I. Tamblyn, J. P. Garrahan, and P. H. Beton, *Phys. Rev. Lett.* **114**, 1 (2015).
 - [9] F. Sciortino, *Eur. Phys. J. B* **64**, 505 (2008).
 - [10] Y. Wang, Y. Wang, D. R. Breed, V. N. Manoharan, L. Feng, A. D. Hollingsworth, M. Weck, and D. J. Pine, *Nature* **491**, 51 (2012).
 - [11] G.-R. Yi, D. J. Pine, and S. Sacanna, *J. Phys.: Condens. Matter* **25**, 193101 (2013).
 - [12] W. Liu, M. Tagawa, H. L. Xin, T. Wang, H. Emamy, H. Li, K. G. Yager, F. W. Starr, A. V. Tkachenko, and O. Gang, *Science* **351**, 582 (2016).
 - [13] A. Striolo, J. Kim, L. Liz-Marzán, L. Tadiello, M. Pauly, C. Murphy, A. Roig, D. Gracias, Y. Xia, J. Reguera, *et al.*, *Faraday discussions* **191**, 117 (2016).
 - [14] J. Zhang, B. A. Grzybowski, and S. Granick, *Langmuir* **33**, 6964 (2017).
 - [15] J. Diaz and D. Pine, *Bulletin of the American Physical Society* (2018).
 - [16] E. Bianchi, R. Blaak, and C. N. Likos, *Physical Chemistry Chemical Physics* **13**, 6397 (2011).
 - [17] J. Tavares, N. Almaraz, and M. T. da Gama, *Soft Matter* **11**, 5828 (2015).
 - [18] J. M. Tavares, C. S. Dias, N. A. M. Arajo, and M. M. Telo da Gama, *J. Phys. Chem. B* **0**, null (0), pMID: 29251935.
 - [19] P. I. C. Teixeira and J. Tavares, *Current Opinion in Colloid & Interface Science* **30**, 16 (2017).
 - [20] Q. Chen, S. C. Bae, and S. Granick, *Nature* **469**, 381 (2011).
 - [21] Y. Iwashita and Y. Kimura, *Soft Matter* **9**, 10694 (2013).
 - [22] S. Biffi, R. Cerbino, F. Bomboi, E. M. Paraboschi, R. Asselta, F. Sciortino, and T. Bellini, *Proc. Nat. Acad. Sci.* **110**, 15633 (2013).
 - [23] F. Bomboi, F. Romano, M. Leo, J. Fernandez-Castanon, R. Cerbino, T. Bellini, F. Bordini, P. Filetici, and F. Sciortino, *Nat. Commun.* **7**, 13191 (2016).
 - [24] W. Liu, N. A. Mahynski, O. Gang, A. Z. Panagiotopoulos, and S. K. Kumar, *ACS nano* **11**, 4950 (2017).
 - [25] N. Kern and D. Frenkel, *J. Chem. Phys.* **118**, 9882 (2003).
 - [26] Zhang, A. S. Keys, T. Chen, and S. C. Glotzer, *Langmuir* **21**, 11547 (2005), pMID: 16316077.
 - [27] E. Bianchi, J. Largo, P. Tartaglia, E. Zaccarelli, and F. Sciortino, *Phys. Rev. Lett.* **97**, 168301 (2006).
 - [28] J. P. K. Doye, A. A. Louis, I.-C. Lin, L. R. Allen, E. G. Noya, A. W. Wilber, H. C. Kok, and R. Lyus, *Phys. Chem. Chem. Phys.* **9**, 2197 (2007).
 - [29] E. G. Noya, C. Vega, J. P. K. Doye, and A. A. Louis, *J. Chem. Phys.* **132**, 234511 (2010).

- [30] E. Bianchi, G. Kahl, and C. N. Likos, *Soft Matter* **7**, 8313 (2011).
- [31] A. B. Pawar and I. Kretzschmar, *Langmuir* **25**, 9057 (2009).
- [32] D. J. Kraft, J. Hilhorst, M. A. P. Heinen, M. J. Hoogenraad, B. Luigjes, and W. K. Kegel, *J. Phys. Chem. B* **115**, 7175 (2011).
- [33] P. D. J. van Oostrum, M. Hejazifar, C. Niedermayer, and E. Reimhult, *J. Phys.: Condens. Matter* **27**, 234105 (2015).
- [34] T. Tigges, D. Hoenders, and A. Walther, *Small* **11**, 4540 (2015).
- [35] S. Roldán-Vargas, F. Smallenburg, W. Kob, and F. Sciortino, *Sci. Rep.* **3**, 2451 (2013).
- [36] L. Rovigatti, V. Bianco, J. M. Tavares, and F. Sciortino, *J. Chem. Phys.* **146**, 041103 (2017).
- [37] F. Smallenburg and F. Sciortino, *Nature Physics* **9**, 554 (2013).
- [38] L. Rovigatti, F. Smallenburg, F. Romano, and F. Sciortino, *ACS Nano* **8**, 3567 (2014).
- [39] S. Biffi, R. Cerbino, G. Nava, F. Bomboi, F. Sciortino, and T. Bellini, *Soft Matter* **11**, 3132 (2015).
- [40] L. Rovigatti, J. Russo, and F. Romano, “PatchyParticles: an educational Monte Carlo code to simulate patchy particles,” (2018).
- [41] F. Sciortino, *Eur. Phys. J. E* **40**, 3 (2017).
- [42] E. Bianchi, P. Tartaglia, E. Zaccarelli, and F. Sciortino, *J. Chem. Phys.* **128**, 144504 (2008).
- [43] L. Rovigatti, D. de las Heras, J. M. Tavares, M. M. Telo da Gama, and F. Sciortino, *J. Chem. Phys.* **138**, 164904 (2013).
- [44] F. Romano, J. Russo, and H. Tanaka, *Phys. Rev. Lett.* **113**, 138303 (2014).
- [45] C. Ggelein, G. Ngele, R. Tuinier, T. Gibaud, A. Stradner, and P. Schurtenberger, *J. Chem. Phys.* **129**, 085102 (2008).
- [46] J. Russo, P. Tartaglia, and F. Sciortino, *J. Chem. Phys.* **131**, 014504 (2009).
- [47] S. Roldan-Vargas, L. Rovigatti, and F. Sciortino, *Soft Matter* **13**, 514 (2017).
- [48] D. J. Audus, F. Starr, and J. Douglas, *Soft Matter*, (2018).
- [49] F. Romano, E. Sanz, and F. Sciortino, *J. Chem. Phys.* **134**, 174502 (2011).
- [50] F. Romano and F. Sciortino, *Soft Matter* **7**, 5799 (2011).
- [51] W. L. Miller and A. Cacciuto, *Phys. Rev. E* **80**, 021404 (2009).
- [52] H. Rezvantlab, D. J. Beltran-Villegas, and R. G. Larson, *Phys. Rev. Lett.* **117**, 128001 (2016).
- [53] N. A. Mahynski, L. Rovigatti, C. N. Likos, and A. Z. Panagiotopoulos, *ACS Nano* **10**, 5459 (2016), pMID: 27124487.
- [54] M. S. Fernández, V. R. Misko, and F. M. Peeters, *Phys. Rev. E* **92**, 042309 (2015).
- [55] I. Coluzza, P. D. van Oostrum, B. Capone, E. Reimhult, and C. Dellago, *Phys. Rev. Lett.* **110**, 075501 (2013).
- [56] F. H. Stillinger and T. A. Weber, *Phys. Rev. B* **31**, 5262 (1985).
- [57] V. Molinero and E. B. Moore, *J. Phys. Chem. B* **113**, 4008 (2008).
- [58] C. Vega and J. L. F. Abascal, *Phys. Chem. Chem. Phys.* **13**, 19663 (2011).
- [59] J. Russo, J. M. Tavares, P. I. C. Teixeira, M. M. Telo da Gama, and F. Sciortino, *Phys. Rev. Lett.* **106**, 085703 (2011).
- [60] R. J. Speedy and P. G. Debenedetti, *Molecular Physics* **81**, 237 (1994).
- [61] E. Zaccarelli, S. V. Buldyrev, E. La Nave, A. J. Moreno, I. Saika-Voivod, F. Sciortino, and P. Tartaglia, *Phys. Rev. Lett.* **94**, 218301 (2005).
- [62] N. Gnan, L. Rovigatti, M. Bergman, and E. Zaccarelli, *Macromolecules* **50**, 8777 (2017).
- [63] L. Rovigatti, G. Nava, T. Bellini, and F. Sciortino, *Macromolecules* **51**, 1232 (0).
- [64] B. Smit and D. Frenkel, *Understanding molecular simulations* (Academic, New York, 1996).
- [65] R. D. Mountain and D. Thirumalai, *Physica A* **210**, 453 (1994).
- [66] D. C. Rapaport, *The art of molecular dynamics simulation* (Cambridge university press, Cambridge, 2004).
- [67] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids* (Oxford university press, New York, 1987).
- [68] M. P. Allen and G. Germano, *Mol. Phys.* **104**, 3225 (2006).
- [69] G. Ciccotti and J. Ryckaert, *Comput. Phys. Rep.* **4**, 346 (1986).
- [70] V. Krutler, W. F. van Gunsteren, and P. H. Hünenberger, *J. Comput. Chem.* **22**, 501 (2001).
- [71] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije, *J. Comput. Chem.* **18**, 1463 (1997).
- [72] A. Dullweber, B. Leimkuhler, and R. McLachlan, *J. Chem. Phys.* **107**, 5840 (1997).
- [73] J. F. Brady and G. Bossis, *Ann. Rev. Fluid Mech.* **20**, 111 (1988).
- [74] R. L. Davidchack, R. Handel, and M. V. Tretyakov, *J. Chem. Phys.* **130**, 234101 (2009).
- [75] R. L. Davidchack, T. E. Ouldridge, and M. V. Tretyakov, *J. Chem. Phys.* **147**, 224103 (2017).
- [76] W. M. Brown, A. Kohlmeyer, S. J. Plimpton, and A. N. Tharrington, *Computer Physics Communications* **183**, 449 (2012).
- [77] M. J. Abraham, T. Murtola, R. Schulz, S. Pll, J. C. Smith, B. Hess, and E. Lindahl, *SoftwareX* **1-2**, 19 (2015).
- [78] N. B. Wilding, *J. Phys.: Condens. Matter* **9**, 585 (1997).
- [79] J. J. Potoff and A. Z. Panagiotopoulos, *J. Chem. Phys.* **109**, 10914 (1998).
- [80] R. L. C. Vink and J. Horbach, *J. Chem. Phys.* **121**, 3253 (2004).
- [81] F. Wang and D. P. Landau, *Phys. Rev. Lett.* **86**, 2050 (2001).
- [82] B. Chen and J. I. Siepmann, *J. Phys. Chem. B* **104**, 8725 (2000).
- [83] B. Chen and J. I. Siepmann, *J. Phys. Chem. B* **105**, 11275 (2001).
- [84] S. Whitelam and P. L. Geissler, *J. Chem. Phys.* **127**, 154101 (2007), <https://doi.org/10.1063/1.2790421>.
- [85] A. Morriss-Andrews, J. Rottler, and S. S. Plotkin, *J. Chem. Phys.* **132**, 035105 (2010), <https://doi.org/10.1063/1.3269994>.
- [86] P. Šulc, F. Romano, T. E. Ouldridge, L. Rovigatti, J. P. K. Doye, and A. A. Louis, *J. Chem. Phys.* **137**, 135101 (2012), <https://doi.org/10.1063/1.4754132>.
- [87] S. Whitelam, E. H. Feng, M. F. Hagan, and P. L. Geissler, *Soft Matter* **5**, 1251 (2009).
- [88] S. Růžička and M. P. Allen, *Phys. Rev. E* **89**, 033307 (2014).
- [89] Z. Zhang and S. C. Glotzer, *Nano Letters* **4**, 1407 (2004).

- [90] S. C. Glotzer and M. J. Solomon, *Nature materials* **6**, 557 (2007).
- [91] F. Sciortino, *Soft Matter Self-Assembly* **193**, 1 (2016).
- [92] F. Sciortino, E. Bianchi, J. F. Douglas, and P. Tartaglia, *J. Chem. Phys.* **126**, 194903 (2007).
- [93] I. Saika-Voivod, F. Romano, and F. Sciortino, *J. Chem. Phys.* **135**, 124506 (2011).
- [94] F. Romano, E. Sanz, P. Tartaglia, and F. Sciortino, *J. Phys.: Condens. Matter* **24**, 064113 (2012).
- [95] D. P. Landau and K. Binder, *A guide to Monte Carlo simulations in statistical physics* (Cambridge university press, 2014).
- [96] P. Virnau and M. Müller, *J. Chem. Phys.* **120**, 10925 (2004).
- [97] L. Filion, M. Marechal, B. van Oorschot, D. Pelt, F. Smalenburg, and M. Dijkstra, *Phys. Rev. Lett.* **103**, 188302 (2009).
- [98] J. de Graaf, R. van Roij, and M. Dijkstra, *Phys. Rev. Lett.* **107**, 155501 (2011).
- [99] M. Marechal, R. J. Kortschot, A. F. Demirrs, A. Imhof, and M. Dijkstra, *Nano Letters* **10**, 1907 (2010), pMID: 20387799.
- [100] T. Vissers, Z. Preisler, F. Smalenburg, M. Dijkstra, and F. Sciortino, *J. Chem. Phys.* **138**, 164505 (2013).
- [101] G. Doppelbauer, E. Bianchi, and G. Kahl, *J. Phys.: Condens. Matter* **22**, 104105 (2010).
- [102] G. Doppelbauer, E. G. Noya, E. Bianchi, and G. Kahl, *Soft Matter* **8**, 7768 (2012).
- [103] E. Bianchi, G. Doppelbauer, L. Filion, M. Dijkstra, and G. Kahl, *J. Chem. Phys.* **136**, 214102 (2012).
- [104] D. Frenkel and A. J. C. Ladd, *J. Chem. Phys.* **81**, 3188 (1984).
- [105] E. G. Noya, M. M. Conde, and C. Vega, *J. Chem. Phys.* **129**, 104704 (2008).
- [106] C. Vega, E. Sanz, J. Abascal, and E. Noya, *J. Phys.: Condens. Matter* **20**, 153101 (2008).
- [107] A. Laio and F. L. Gervasio, *Reports on Progress in Physics* **71**, 126601 (2008).
- [108] A. Barducci, G. Bussi, and M. Parrinello, *Phys. Rev. Lett.* **100**, 020603 (2008).
- [109] G. M. Torrie and J. P. Valleau, *J. Comput. Phys.* **23**, 187 (1977).
- [110] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Vol. 1 (Elsevier, 2001).
- [111] A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* **61**, 2635 (1988).
- [112] S. Prestipino, *J. Chem. Phys.* **148**, 124505 (2018).
- [113] D. Richard and T. Speck, *J. Chem. Phys.* **148**, 124110 (2018), <https://doi.org/10.1063/1.5016277>.
- [114] J. Russo, F. Romano, and H. Tanaka, *Nature materials* **13**, 733 (2014).
- [115] L. Rovigatti, J. Russo, and F. Sciortino, *Phys. Rev. Lett.* **107**, 237801 (2011).
- [116] K. T. Nguyen, F. Sciortino, and C. De Michele, *Langmuir* **30**, 4814 (2014), pMID: 24701976.